NODE.JS NIGHTS

STRV

SECURITY BASICS

Jirka Erhart, Backend developer at STRV

STRV

PRINCIPLES

OWASP

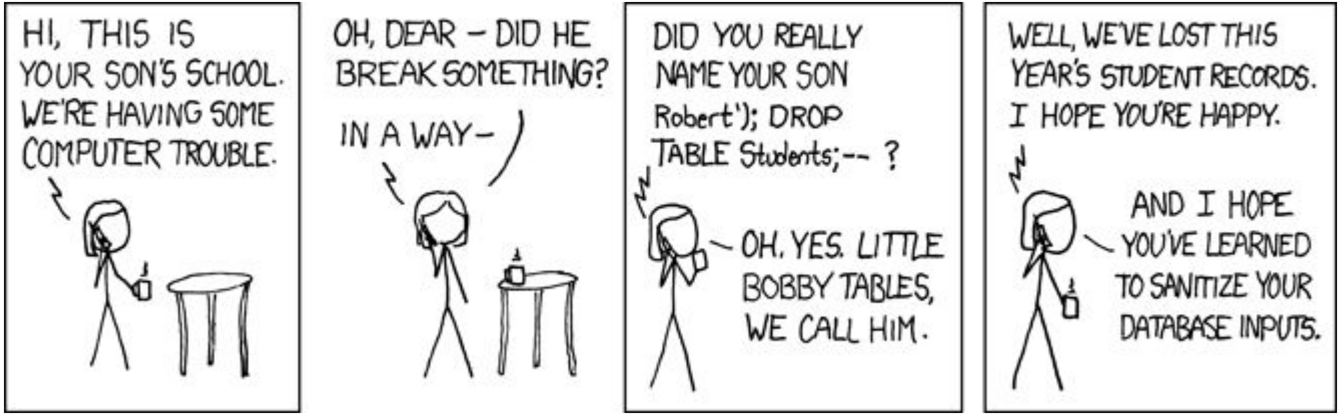
- Open Web Application Security Project
- Top 10 project
- www.owasp.org

RESOURCE ACCESS

- Always check access privileges
- Return least data amount needed
- Whitelist returned data

SQL INJECTION

- Require strict data input
- Use ORM
- Sanitize input (even in raw queries!)


HI, THIS IS YOUR SON'S SCHOOL. WE'RE HAVING SOME COMPUTER TROUBLE.
OH, DEAR – DID HE BREAK SOMETHING?
IN A WAY–
DID YOU REALLY NAME YOUR SON Robert'); DROP TABLE Students;-- ?
OH. YES. LITTLE BOBBY TABLES, WE CALL HIM.
WELL, WE'VE LOST THIS YEAR'S STUDENT RECORDS. I HOPE YOU'RE HAPPY.
AND I HOPE YOU'VE LEARNED TO SANITIZE YOUR DATABASE INPUTS.

X-SITE SCRIPTING

- Malicious script stored in database and then loaded on client via API
- Sanitize input before saving
- Client should always display as text and not as HTML

AUTHENTICATION TOKENS

- Don't use endless tokens (always add expiration)
- Use refresh tokens
- Have a way to deny user access
- (Or use stateful authorization)

LOGGING

- Never log sensitive data
    - Passwords
    - Access tokens
    - Social Security Numbers
    - ...

STORING DATA

- Always Hash passwords (with salt)
- Encrypt SSN, bank info and other sensitive data

FE - BE CODE SHARING

- Never share confidential business logic on frontend
- Never share code doing back-end encryption/decryption on frontend
- Always check which environment variables are being exposed in frontend bundle

INPUT VALIDATION

- Validate all incoming data
- Be as strict as possible
- All strings should always have **maximum length defined**
- Filters should limit maximum number of results

KEYS AND PASSWORDS

- NEVER EVER COMMIT THEM TO GITHUB (they will remain in history)
- Store them in environment variables
- Never send/show them
- Reduce amount of people having access to production keys/passwords
    - Good for project security
    - Protects people (when something happen they are not on the suspects list)

DATABASE ACCESS

- Enforce SSL and authentication
- Always set up automated backups

THIRD PARTY SERVICES

- Enforce authenticated access (when you exchange private info)
- Enforce SSL

CONFIGURATION KEY POLICY

- Min. 32 chars
- Use special character, numeric and upper and lower case letters
- Never send passwords/keys via email/slack, etc. (split them)

TOOLS

HELMET

- Takes care of your headers
- Prevents some XSS vulnerabilities
- https://www.npmjs.com/package/koa-helmet

NPM AUDIT

- Checks for common vulnerabilities in your library tree
- Checks against npm repository data
- Has a fix feature
- https://docs.npmjs.com/cli/audit

SNYK

- Finds and fixes library vulnerabilities
- Comprehensible vulnerability database
- Provides patches as pull requests
- Integrates easily
- https://snyk.io/

THAT'S IT

Jirka Erhart
jiri.erhart@strv.com

STRV

QUESTIONS

STRV